**Кратко:**

Обучающая алгоритмизации игра для детей 6-10 лет, подготавливающая к изучению программирования. Написана под Android на движке Unity с использованием C#. Выполнена как 2d-игра в изометрической проекции, имеет 12 уровней, сюжет, анимации движений персонажа и объектов, меню выбора уровней, игровое поле, меню программирования действий игрового персонажа. Все игровые модели и объекты не заимствованы из интернета, а спроектированы мною лично. Это приложение с закрытым исходным кодом!

**Требования:**

1. Приложение должно быть работоспособным на платформах Android версий 4.2 - 6.0, как наиболее распространенных;
2. Иметь яркий и привлекательный для детей интерфейс с оптимальным размером кнопок, с музыкой и звуками;
3. Содержать сцены: выбора уровня, игровую, с панелью ввода команд (без ограничения количества действий). Написанный алгоритм, а также результаты его выполнения должны сохраняться внутри игры;
4. Иметь тематику игры и мини-сюжет: восстановление экологии планеты после экологической катастрофы, действующий персонаж – робот, с кодовым именем «КоробОК»;
5. Иметь прямоугольное игровое поле размером 6×6 клеток, причем один игровой объект занимает ровно одну клетку;
6. Игровая сцена должна быть выполнена в 2D графике с видом от третьего лица, в изометрической проекции, для наилучшего соотношения реалистичности и производительности игры и простоты разработки;
7. Иметь максимально простую и наглядную систему команд робота (робот может поворачиваться влево и вправо, а также перемещаться вперёд и назад) и содержать все базовые алгоритмические конструкции (линейная, ветвление и циклы);
8. Определять баллы за прохождение уровня, в зависимости от количества команд в программе и использования для прохождения алгоритмической конструкции, если это предполагается в конкретном уровне;
9. Содержать 12 изначально открытых уровней разной сложности, доступных для детей 6-9 лет. По 4 уровня на каждую алгоритмическую структуру (для альфа-версии).

Суть игры заключается в том, что робот должен, по написанному учеником алгоритму движения, доставлять ящики с растениями на плодородную почву. На его пути может оказаться радиация. Для преодоления этого препятствия необходимо использовать условную алгоритмическую конструкцию. В некоторых уровнях роботу необходимо выполнять повторяющиеся действия, что предполагает использованием циклов, при построении алгоритма.

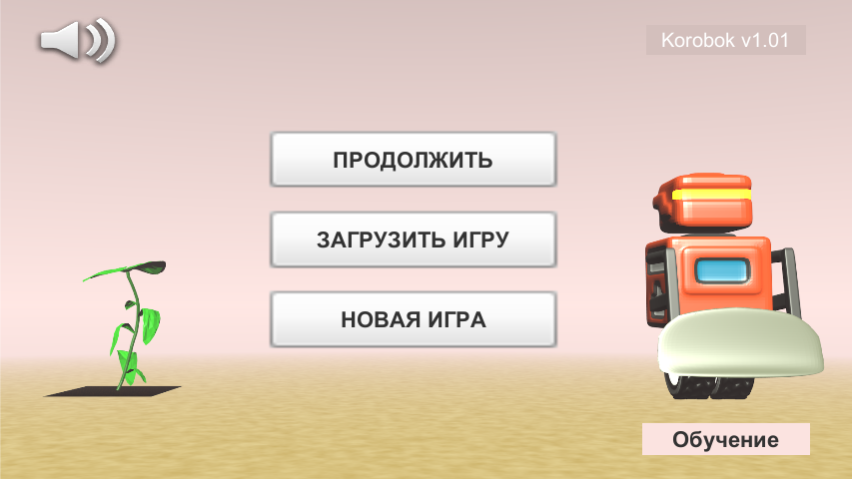
****

Рисунок 11. Стартовая сцена игры

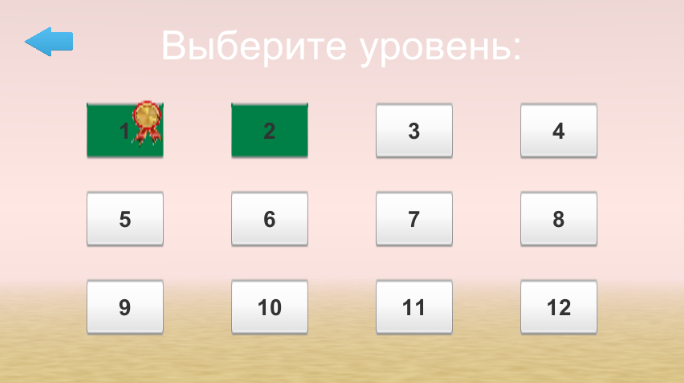
****

Рисунок 12. Сцена выбора уровня

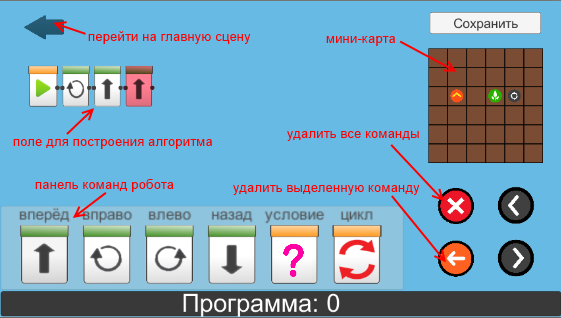
****

Рисунок 13. Сцена составления алгоритма для игрового персонажа

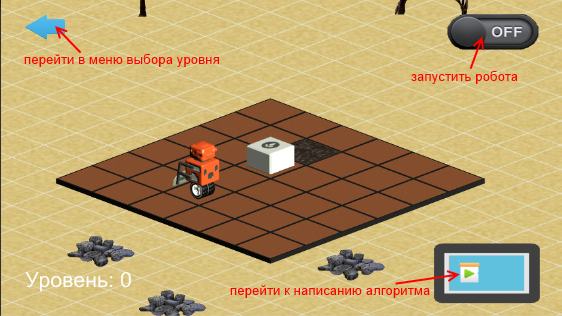
****

Рисунок 14. Основная сцена игры

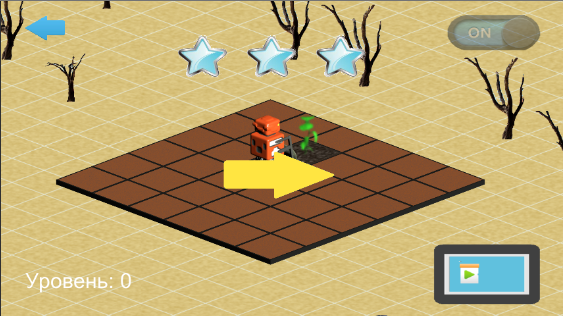
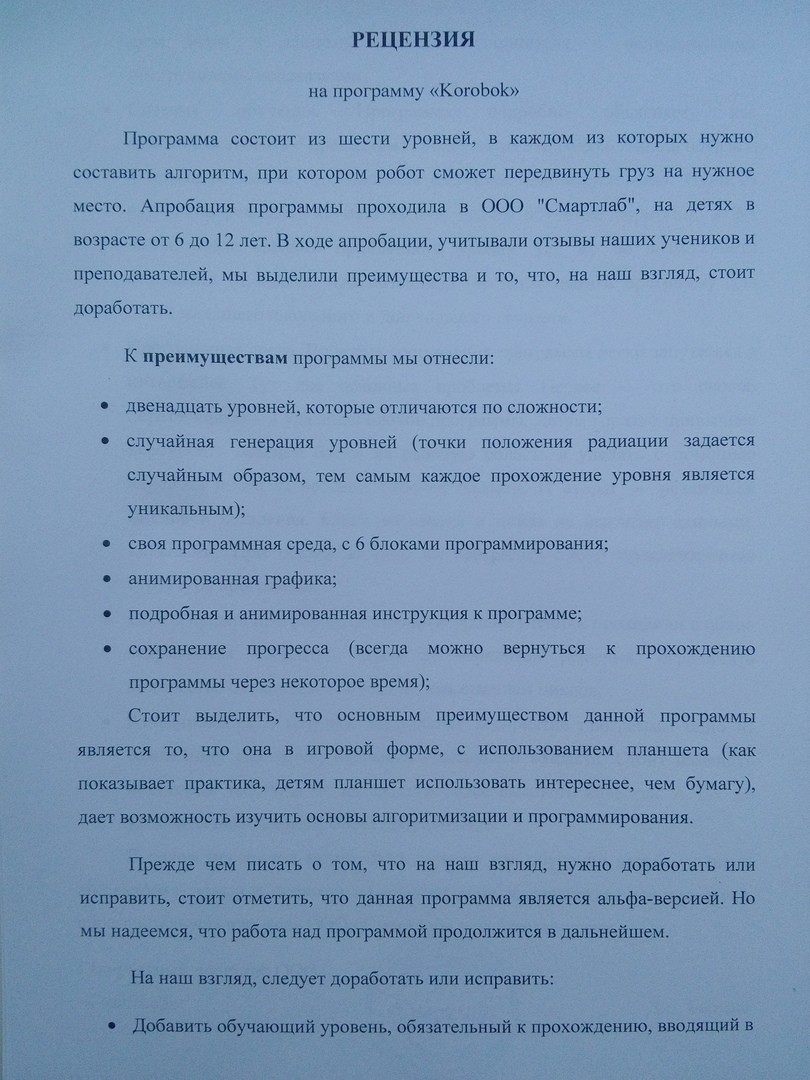
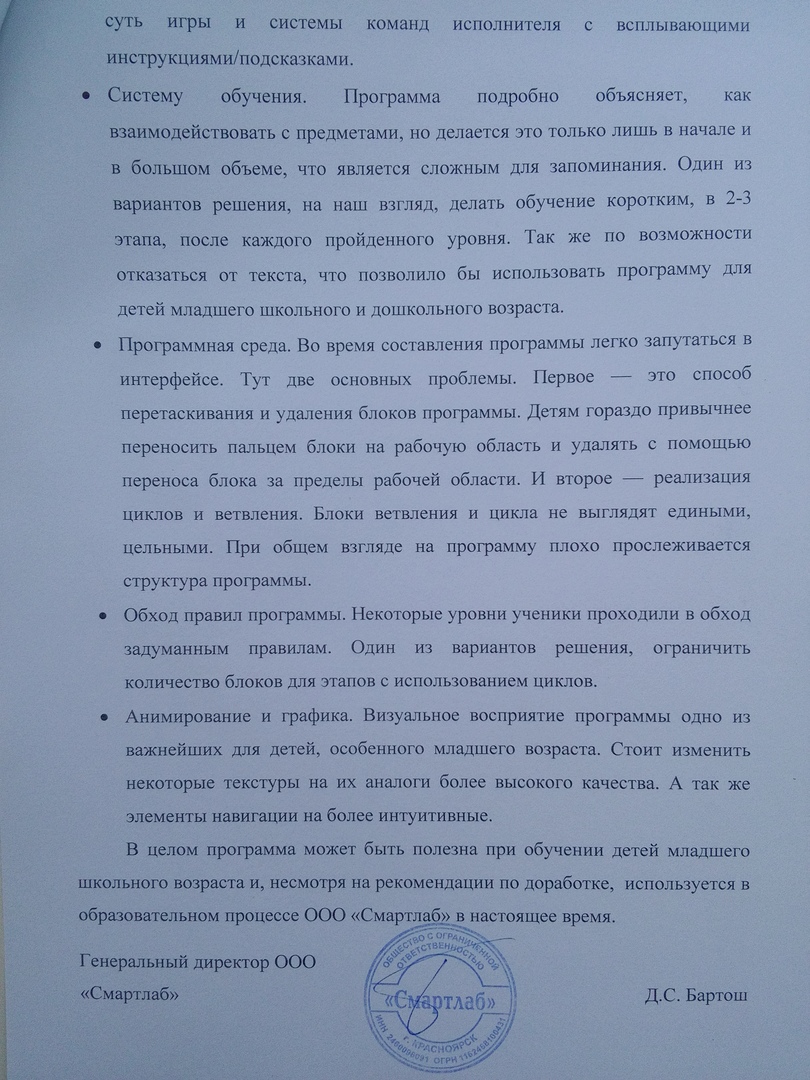


Рисунок 15. Успешное прохождение уровня

**Игра включает в себя:**

1. Несколько открытых изначально игровых миссий и мини-сюжет (по сюжету на Земле, из-за загрязнения окружающей среды, вымерли все растения и роботу необходимо доставлять ящики с растениями на плодородную почву);
2. Стартовое меню, где можно загрузить старую или начать новую игру, а также отключить звук или запустить обучение (см. рис 11);
3. Меню выбора уровня, в котором пройденные уровни помечаются зеленым цветом, а уровни, пройденные с эффективно составленным алгоритмом, помечаются медалькой (см. рис 12);
4. Сцену с игровым полем, по которому робот перемещается, выполняя заданные игроком команды (см. рис 13);
5. Простую и наглядную систему команд игрового персонажа и все базовые алгоритмические конструкции (см. рис. 14);
6. Систему оценивания эффективности написанной игроком программы, после прохождения уровня даются звёздочки (см. рис. 15).





Данную игру рекомендуется использовать перед началом изучения программирования на занятиях по робототехнике и на уроках по информатике. Также игру «Коробок» можно использовать детям для развлечения и родителям для организации полезного досуга детей. Игра предназначена для детей младшего школьного возраста, однако, с не меньшим интересом и пользой в нее играют дети и более старшего возраста.

Приложение включает в себя 12 уровней на три алгоритмических конструкции. Первые 4 уровня – это первый этап, где для успешного прохождения используется только линейная конструкция. Уровни с 5 по 8 включительно – это второй этап, где необходимо использовать ветвление. Последний третий этап (уровни с 9 по 12) предполагает обязательное использование циклов. Для удобства использования игры на занятиях – все уровни изначально открыты. Уровни, предусматривающие использование ветвления и циклы можно пройти и без использования этих конструкций. Но для получения максимального количества баллов – их использование обязательно.

Во избежание путаницы в сцене построения алгоритма запрещено использовать ветвление внутри ветвления и цикл в цикле. Но можно использовать цикл в ветвлении и наоборот. Чтобы не ограничивать свободу мышления ребёнка, изначально открыты все команды робота и все конструкции, ведь каждый уровень можно пройти несколькими разными алгоритмами.

Уровни второго и третьего этапа можно сначала пройти ученику без использования ветвлений и циклов, а затем показать целесообразное их использование. Ведь, если проходить уровни первого этапа без ветвления, то алгоритм не всегда будет рабочим, так как радиация появляется в случайной клетке. А если проходить уровни второго этапа без циклов, то придётся писать большое количество повторяющихся команд.

Если ребёнок уже знаком с основными алгоритмическими конструкциями, то можно эту игру дать ему на самостоятельное прохождение. Возможна организация прохождения этапов в группах по 2-3 человека.

Рекомендуется разделить прохождение игры по алгоритмическим конструкциям на три занятия. Порядок прохождения уровней внутри каждого этапа не имеет значения, однако, порядок прохождения этапов менять не следует.

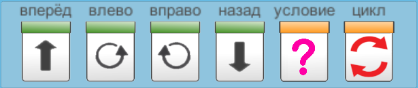


Рисунок 16. Панель команд Коробка

**Первое занятие: Линейные алгоритмы**

Преобладающим методом обучения на занятии будет являться игра, поскольку использование игрового приложения предполагает использование данного метода. Но в начале урока потребуется использовать объяснение. При объяснении – учитель излагает строго и последовательно основные и необходимые для данного этапа правила игры, а также знакомит ученика с её функционал и интерфейсом.

На первом занятии рекомендуется просмотр обучающего ролика, который предлагается в начале игры. Также проводиться ознакомление ученика с основными командами игрового персонажа.

*Правила, которые должен знать ученик перед началом игры*

Игровой персонаж – робот с кодовым именем «КоробОК» должен доставлять ящики с растениями на плодородную почву. Он может перемещаться по игровому полю, которое имеет размер 6×6 клеток, вперёд и назад, а также поворачиваться вправо и влево. На его пути может встретиться препятствия в виде каменных завалов. По таким клеткам робот двигаться не может. Ящик с растением он может толкать только ковшом, то есть только когда движется вперёд. Он не может толкать два ящика сразу или толкать его при движении назад.

Для того чтобы робот выполнил свою миссию, необходимо в отдельной игровой сцене составить для него линейный алгоритм. Алгоритмом в данном случае будет набор действий робота, выбранных ребёнком. Линейный алгоритм – это когда каждое действие выполняется по очереди друг за другом. Каждой команде робота соответствует кнопка на панели команд, которую необходимо нажимать, чтобы записать команду в алгоритм (см. рис. 16). Команды можно удалять отдельно или удалить все сразу с помощью кнопок под мини-картой.

*Краткое содержание этапа*

Для удобства в этой сцене отображается мини-карта. На рисунке 17 представлены мини-карты для первых четырёх игровых уровней. Ученик составляет алгоритм и пробует запустить его.



Рисунок 17. Мини-карты уровней первого этапа

После каждой попытки прохождения уровня рекомендуется проводить рефлексию и анализ алгоритма, почему, например, данный алгоритм оказался неудачным, или почему он не был оценен на три звезды. Если у ребёнка совсем не получается пройти ни один из уровней, то можно попробовать показать ему правильный алгоритм прохождения одного из уровней этого этапа. Либо демонстрировать выполнение персонажем различных алгоритмов, не обязательно выполняющих основную задачу (доставить ящик на плодородную почву).

**Второе занятие: Ветвление**

На втором занятие будут использоваться те же методы обучения, что и на первом: объяснение, игра, решение задач.

*Правила, которые должен знать ученик перед началом игры*

Во-первых, необходимо повторить правила из первого занятия. Во-вторых, необходимо объяснить правила связанные с неизвестными клетками поля и радиацией. Ровно половина клеток, отмеченных вопросительным знаком, заражена радиацией. Робот сканирует клетку, которая находится впереди него. Какая из клеток заражена не известно заранее (так как это определяется случайным образом в момент запуска алгоритма). Через радиацию робот не может ехать и толкать ящики, потому что она губительна и для него и для растения. Во избежание уничтожения растения, роботу запрещено толкать ящик через неизвестные клетки, сначала эти клетки необходимо просканировать.

Чтобы добавить в алгоритм ветвление (можно называть его просто «ветка»), нужно нажать на кнопку с вопросительным знаком на панели команд персонажа (см. рис. 16). Вставлять «ветку» нужно в том месте алгоритма, где робот будет стоять перед клеткой с вопросительным знаком. Теперь алгоритм разветвился. Внизу пишутся команды для случая, если робот не обнаружил на впереди идущей клетке радиации. Сверху добавляются действия, которые робот должен выполнить, если радиация все-таки есть.



Рисунок 18. Мини-карты уровней второго этапа

*Краткое содержание этапа*

Задания для робота на втором этапе изображены на рисунке 18. После повторения правил, рекомендуется объяснить ребёнку, что использование Ветвления сделает возможным прохождение уровня с одним и тем же алгоритмом, вне зависимости от того, в каком месте окажется радиация. Именно для этого используется ветвление. Для лучшего понимания, можно многократно запустить, написанный учителем алгоритм.

**Третье занятие: Циклы**

На третьем занятии необходимо сформировать представление о циклической алгоритмической конструкции. Методы обучения: объяснение, игра, решение задач.

*Правила, которые должен знать ученик перед началом игры*

Циклы нужны затем, чтобы не записывать повторяющиеся действия в алгоритм, а просто указать сколько раз нужно их повторить. Например, на рисунке 21 среди заданий для третьего этапа в 11 уровне робот, для успешного прохождения должен повернуться влево, проехать вперед, повернуться вправо, проехать вперёд и сделать это он должен ровно 4 раза. То есть алгоритм станет в 4 раза меньше по объему, благодаря циклу. Можно даже наглядно продемонстрировать оба алгоритма и, что они оба рабочие. Однако алгоритм без цикла не получит 3 звезды.

Чтобы добавить в алгоритм цикл, нужно нажать на кнопку с двумя круговыми стрелочками на панели команд персонажа (см. рис. 16). Необходимо сначала вставить цикл, а уже внутри него вставлять повторяющиеся команды. После составления набора команд, необходимо указать сколько раз их нужно повторить. Для этого нужно нажать на левый блок цикла с цифрой (она и означает, сколько раз будут повторяться команды) и выбрать другую цифру от 1 до 8. Теперь алгоритм зациклился.

*Краткое содержание этапа*

На рисунке 19 изображены задания для третьего этапа. После прохождения учеником этапа, обязательно обсудить, можно ли использовать эту алгоритмическую структуру в предыдущих уровнях. Если ребёнок справился с заданиями быстро, то можно попробовать пройти с ним какой-нибудь из предыдущих уровней с наименьшим количеством команд.

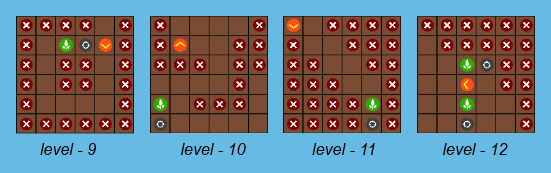


Рисунок 19. Мини-карты уровней третьего этапа